STARDUST meeting - September 12th-13th 2022 @ Kent

A model of actors and grey failures

Laura Bocchi, Simon Thompson, Laura Voinea
Julien Lange

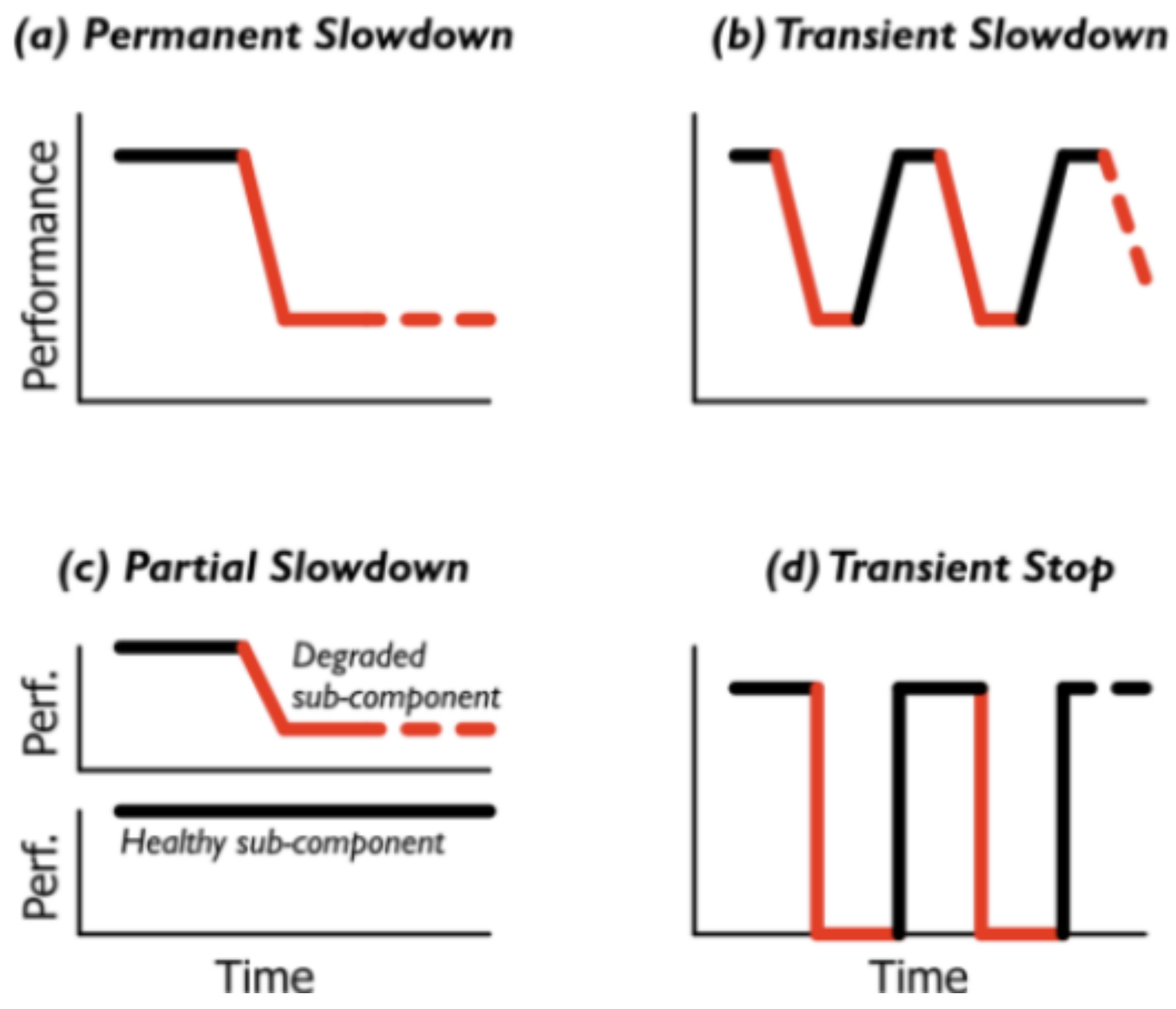
University of Kent
Royal Holloway, University of London

Context

- Software execution is affected by **unpredictability**
- Programming practice: handling unpredictability with timeouts, exceptions, supervisors, …
- Formal models:
  - include reliability primitives e.g., exceptions [Fowler et al. POPL'19], timeouts [Laneve, Zavattaro, FoSSaCS'05][Lopez, Perez, WS-FM'11] …
  - include link/node failures [Francalanza, Hennessy, CONCUR'05] [Adameir, Peters, Nestmann, FORTE'17]…
  - …
- Models mostly focus on fail-stop failures

Grey failures

Fail stop | Grey | Byzantine

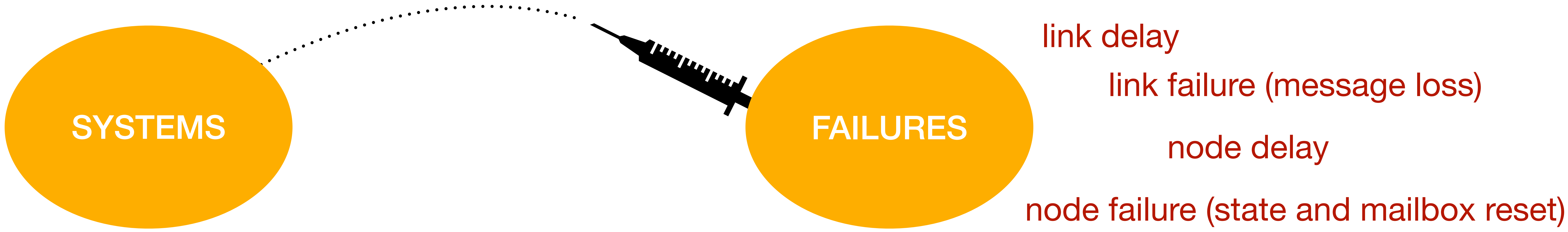


- Multiple “patterns” of failure
- No precise awareness of the state of health of the system
  - A component appears to be working but is experiencing issues
  - Differential observation
- Diagnosis is challenging

[Gunavi et al. ACM Transactions on Storage’18]

Contribution

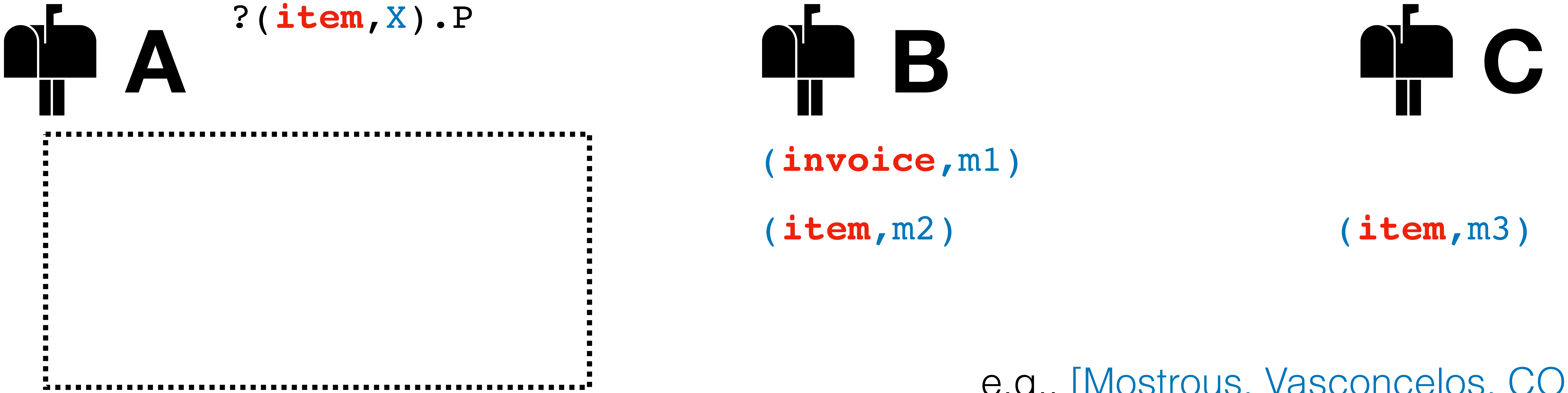
- A formal framework to study the ability of a system to cope with failures
    - a formal model of **actors** and **grey failures**



- a definition of **resilience** and **recoverability**, based on behavioural equivalence

Systems: the main ingredients

- Actor-based systems : we borrow **mailboxes** + **timeouts**
- Mailboxes ➜ expressive communication model e.g., many producers-one consumer
  - asynchrony + pattern-matching + selective receive



e.g., [Mostrous, Vasconcelos, COORDINATION'11]

Systems

$$R ::= \mathtt{n}[P](M)(t) \mid R \parallel R \mid 0$$

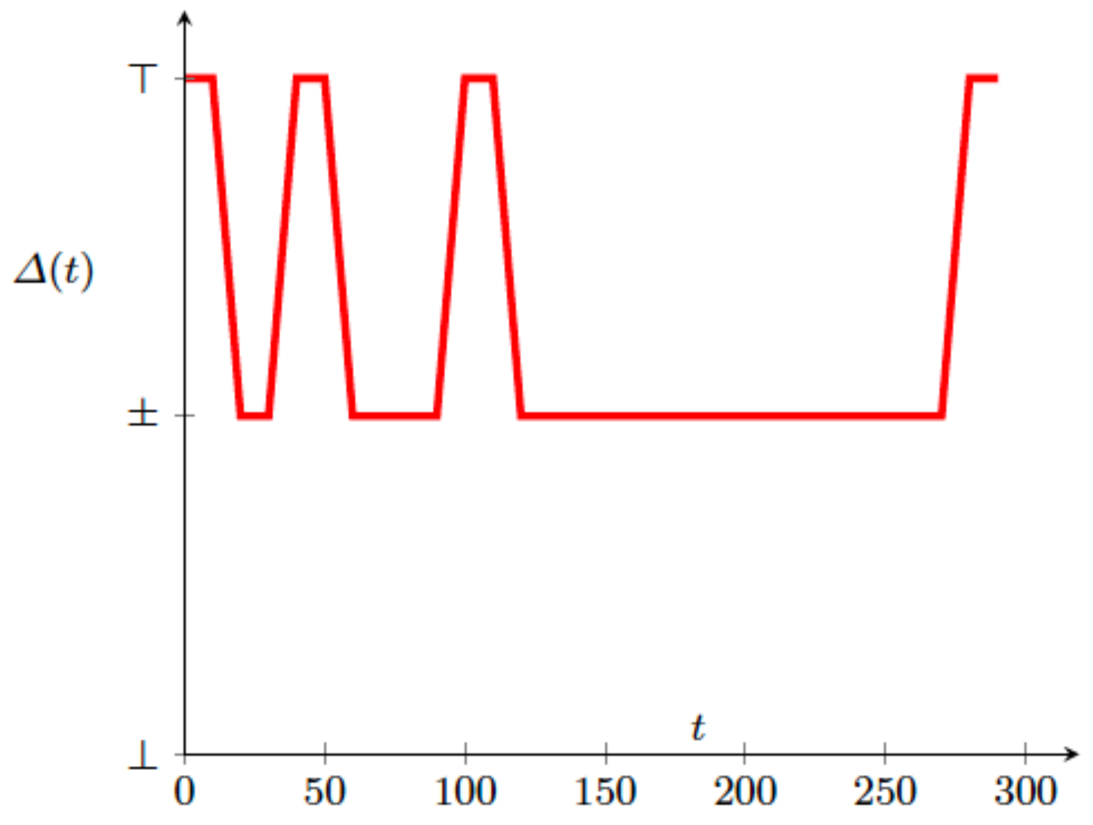
$$P ::= ?\{\mathtt{p}_i\,.\,P_i\}_{i\in I}\ \mathtt{after}\ P \mid !\{\mathtt{n}_i m_i\,.\,P_i\}_{i\in I} \mid \mathtt{sleep}\,.\,P \mid \mu\mathtt{t}\,.\,P \mid \mathtt{t} \mid 0$$

```
receive
  pattern1 -> P1;
  …
  patternN -> PN
after
  n -> P
end
```

Failures

$$\Delta : \mathbb{N} \times \mathcal{N} \cup (\mathcal{N} \times \mathcal{N}) \mapsto \{\top, \bot, \pm\}$$

$$\Delta(\mathtt{n})(t) = \begin{cases} \top & \text{if } t = n^2\ (n \in \mathbb{N}) \\ \bot & \text{otherwise} \end{cases}$$



Producer - consumer

instantaneous $\rightarrow$ time $\rightsquigarrow$ average network latency = 1

```
    p [ sleep. !c item. 0 ](0) || c [ ?item -> 0 after 3 Cf ](0)                time 0
⇝   p [ !c item. 0 ](0) || c [ ?item -> 0 after 2 Cf ](0)                       time 1
→   p [ 0 ](0) || c [ ?item -> 0 after 2 Cf ](0) || sleep.(c,p,item)
⇝   p [ 0 ](0) || c [ ?item -> 0 after 1 Cf ](0) || (c,p,item)                  time 2
→   p [ 0 ](0) || c [ ?item -> 0 after 1 Cf ](item)
→   p [ 0 ](0) || c[ 0 ](0)
```

Producer - consumer with link slowdown

- Reduction on (R, Δ)

$$\Delta(\mathrm{p}, \mathrm{c})(t) = \begin{cases} \pm & \text{if } t \in \{1\} \\ \top & \text{otherwise} \end{cases}$$

$R =$ `p [ sleep. !c item. 0 ](0) || c [ ?item -> 0 after 3 Cf ](0)` time 0

$\rightsquigarrow$ `p [ !c item. 0 ](0) || c [ ?item -> 0 after 2 Cf ](0)` time 1

$\rightarrow$ `p [ 0 ](0) || c [ ?item -> 0 after 2 Cf ](0) || sleep.(p,c,item)` (p,c)

$\rightsquigarrow$ `p [ 0 ](0) || c [ ?item -> 0 after 1 Cf ](0) || sleep.(p,c,item)` time 2

$\rightsquigarrow$ `p [ 0 ](0) || c [ Cf ](0) || (p,c,item)` time 3

Behavioural equivalence

Time-abstract weak barbed bisimulation:

$(R_1, \Delta_1) \approx (R_2, \Delta_2)$ implies:

1) If $(R_1, \Delta_1) \to (R_1', \Delta_1)$ then $(R_2, \Delta_2) \to^* (R_2', \Delta_2)$ and $(R_1', \Delta_1) \approx (R_2', \Delta_2)$
2) If $R_1 \downarrow x$ then $(R_2, \Delta_2) \to^* (R_2', \Delta_2)$ and $R_2' \downarrow x$

Barbs:

$\mathtt{n}[!\{\mathtt{n}_i\, m_i\,.\,P_i\}_{i\in I}](M)(t) \downarrow\ !\mathtt{n}_i\, m_i$ for all $i \in I$

$(\mathtt{n}_1, \mathtt{n}_2, m)(t) \downarrow\ !\mathtt{n}_2\, m$

$\mathtt{n}[?\{\mathtt{p}_i\,.\,P_i\}_{i\in i}\ \mathtt{after}\ P](M)(t) \downarrow\ ?\mathtt{n}\,\mathtt{p}_i$ for all $i \in I$

Example

$R =$ `p[sleep.!c item.0](0) || c[?item->0 after 3 0](0)` $\quad (R, \Delta) \not\approx (R, \top)$

$(R, \top) \quad \to^*$ `p [0](0) || c [0](0)`

$(R, \Delta) \quad \to^*$ `p [0](0) || (p,c,item) || c[0](0)`

Resilience

- Bisimulation to check the ability of a system to preserve behaviour despite failures
- An initial cursed system (R, Δ) is **resilient** if $(R, \Delta) \approx (R, \top)$
- `p[sleep.!c item.0](0) || c[?item->0 after 3 0](0)` is **not resilient** (wrt Δ)

$R_1 =$ `p[sleep.!c item.0](0) || c[?item->0 after 4 0](0)` $\quad (R_1, \Delta) \approx (R_1, \top)$

$R_2 =$ `p [sleep.!c item.0](0) || c[?item->0 after 3 (?item->0 after 3 0)](0)`

$(R_2, \Delta) \approx (R_2, \top)$

$R_3 =$ `p [`μ`t.sleep.!c item.t](0) || p'[`μ`t.sleep.!c item.t](0)` $\quad (R_3, \Delta) \approx (R_3, \top)$
`|| c[`μ`t.?item -> 0 after 3 t](0)`

Recoverability

- Resilience is too strong to capture e.g., retry strategies

```
p[μt.sleep.!c item.?{ok.0, retry.t}](0) ||
    c[μt.?item->!p ok.0 after 3.!p retry.t](0)
```

- An initial cursed system (R, Δ) is ***n*-recoverable** if there exists R' such that $(R, \Delta) \rightarrow^* (R', \Delta)$, time$(R') = n$, and $(R, \top) \approx (R', \Delta)$

Finite thanks to a non-zenoness requirement

Augmentations

- We want to assess resilience/recoverability when **improving** it with recovery strategies

$R =$ `p[sleep.!c item.0](0) || c[?item->0 after 3 0](0)` not resilient wrt Δ

$R_2 =$ `p[sleep.!c item.0](0) ||`
`c[?item->0 after 3 (?item->0 after 3 0)](0)` resilient wrt Δ

- $\overline{R}$ is an augmentation of R if
  (**transparency**) $(\overline{R}, \top) \approx (R, \top)$
  (**improvement**) There exists *n* and Δ s.t. $(\overline{R}, \Delta)$ is n-recoverable and (R, Δ) is not

- Moreover, $\overline{R}$ is **preserving** if for all *n* and Δ,
  (R, Δ) is n-recoverable implies $(\overline{R}, \Delta)$ is *n*-recoverable

Hiding

- Augmentations may add actions e.g., circuit breakers, … so how about hiding?
- Hiding nodes is not enough (too coarse)

Scoped barbs:

- N is a finite set of elements $!\mathtt{n}\,p$ or $?\mathtt{n}\,p$
- $R \downarrow_N ?\mathtt{n}\,p$ if $R \downarrow ?\mathtt{n}\,p$ and $?\mathtt{n}\,p \notin N$
- $R \downarrow_N ?\mathtt{n}\,m$ if $R \downarrow ?\mathtt{n}\,p$ implies $!\mathtt{n}\,p \nvdash_{\text{match}} !\mathtt{n}\,m \quad \forall !\mathtt{n}\,p \in N$

A characterisation of failures

- Fundamentals of fault-tolerant distributed computing in asynchronous environments [Gartner 99]
- Simulation relations for fault-tolerance [Demasi, Castro, Maibaum, Aguirre]
- Fault-tolerance:
  - masking (safety+liveness) : $(R, \Delta) \approx (R, \top)$
  - fail-safe (safety) : $(R, \Delta) \lesssim (R, \top)$
  - non-masking (liveness) : $(R, \Delta) \gtrsim (R, \top)$, n-recoverability, …

Conclusion & future work

- A model for studying **grey failures** in **actor models** (**mailboxes** + **timeouts**)
- Evaluation of recovery strategies reduced to a bisimulation problem
  - resilience, n-recoverability as time-abstract weak barbed bisimulation
  - also augmentation but with existential & universal quantification on n and Δ
- Open questions
  - automated checking and choice of test suites of failures
  - relationships with session types and use in subtyping

Thank you!